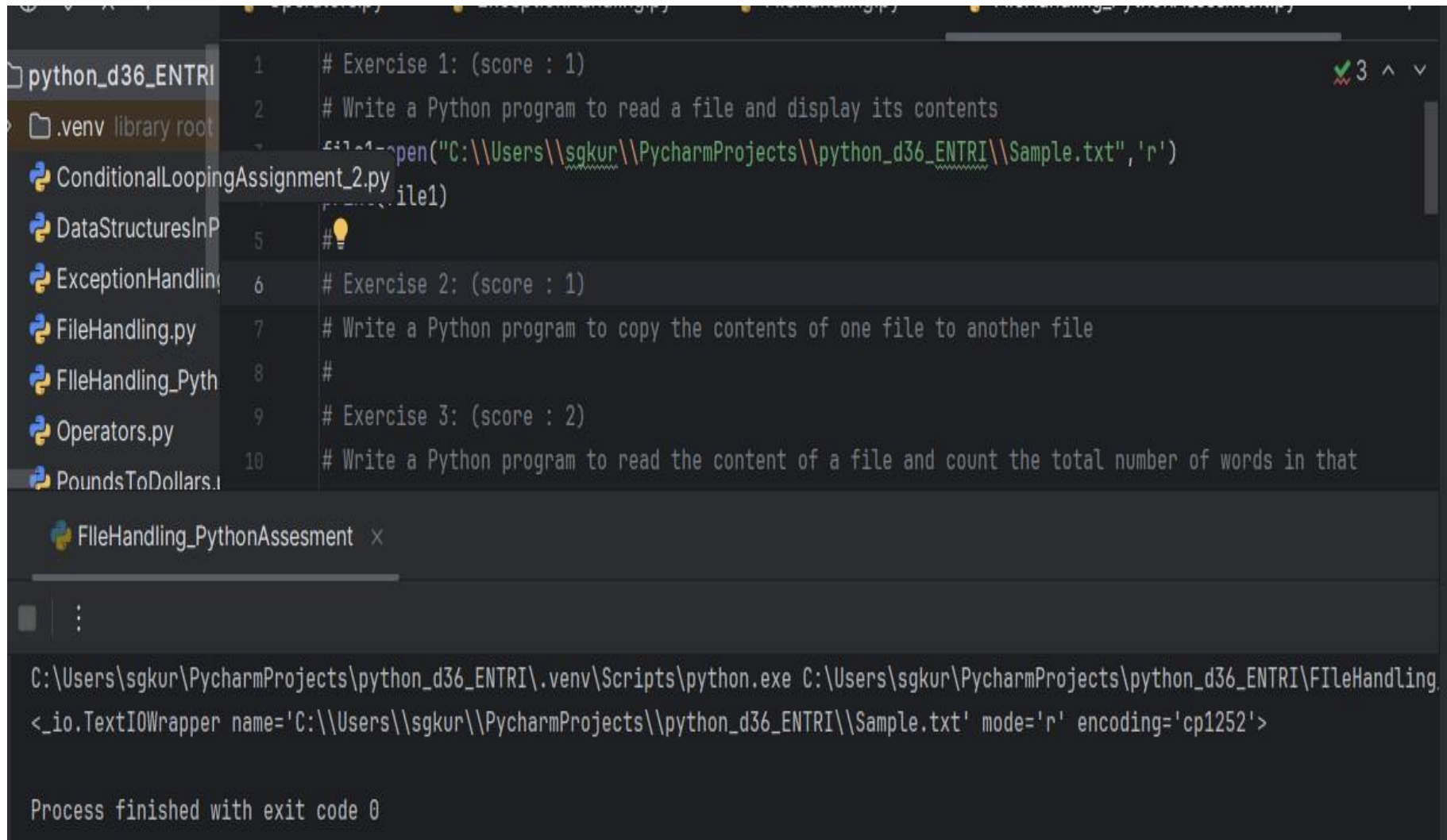


ASSIGNMENT ON FILE AND EXCEPTION HANDLING

By
Shino Mary Philipose

Q.1



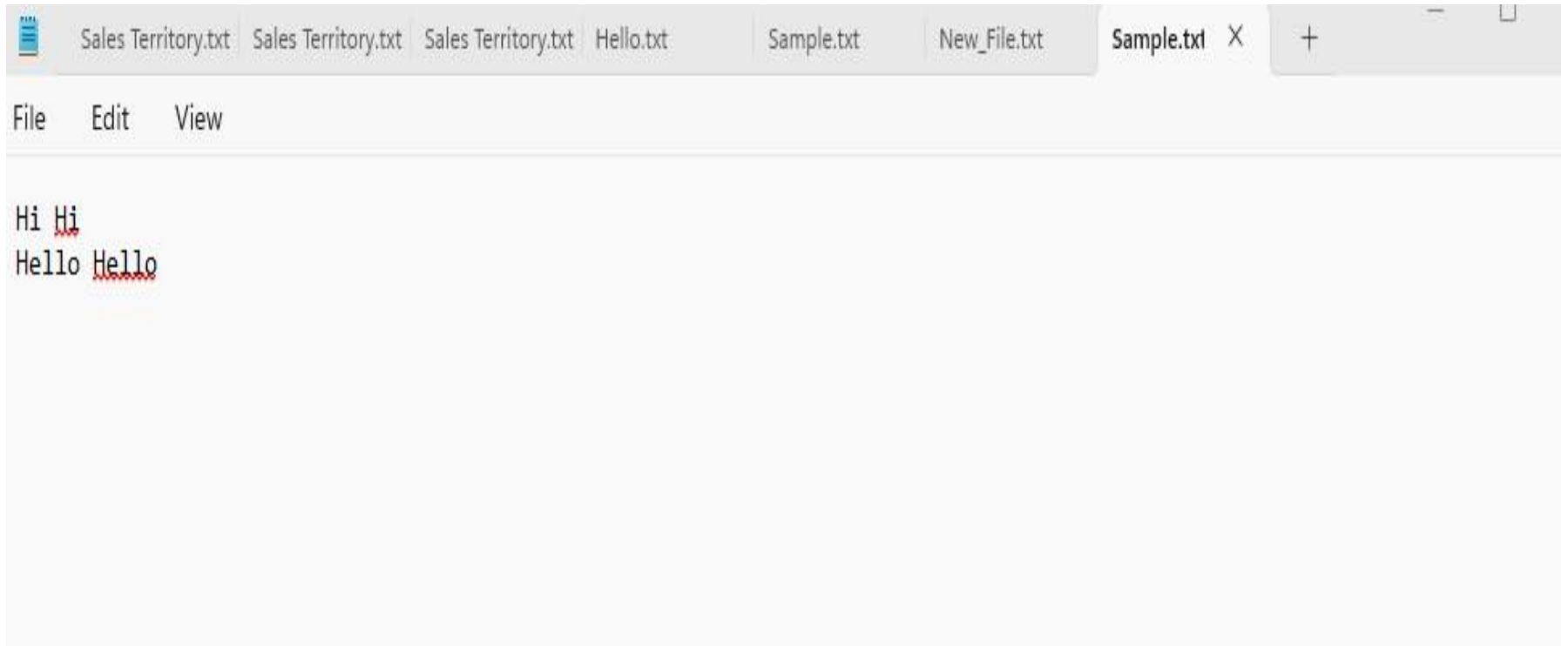
```
python_d36_ENTRI 1 # Exercise 1: (score : 1)
                  2 # Write a Python program to read a file and display its contents
                  3 f=open("C:\\Users\\sgkur\\PycharmProjects\\python_d36_ENTRI\\Sample.txt",'r')
                  4 f.read()
                  5 #
                  6 # Exercise 2: (score : 1)
                  7 # Write a Python program to copy the contents of one file to another file
                  8 #
                  9 # Exercise 3: (score : 2)
                  10 # Write a Python program to read the content of a file and count the total number of words in that

FileHandling_PythonAssesment x

C:\Users\sgkur\PycharmProjects\python_d36_ENTRI\.venv\Scripts\python.exe C:\Users\sgkur\PycharmProjects\python_d36_ENTRI\FileHandling_PythonAssesment.py
<_io.TextIOWrapper name='C:\\Users\\sgkur\\PycharmProjects\\python_d36_ENTRI\\Sample.txt' mode='r' encoding='cp1252'>

Process finished with exit code 0
```

Q.1

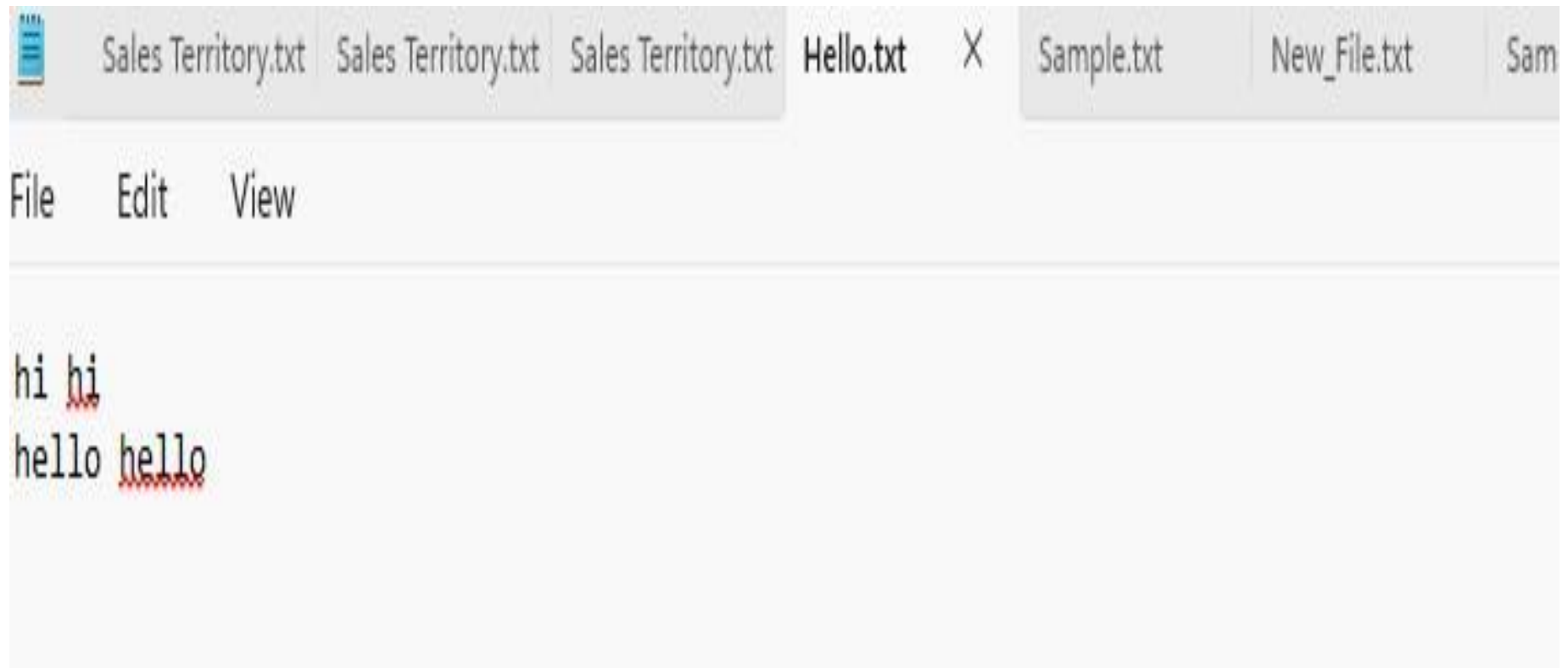


Q.2

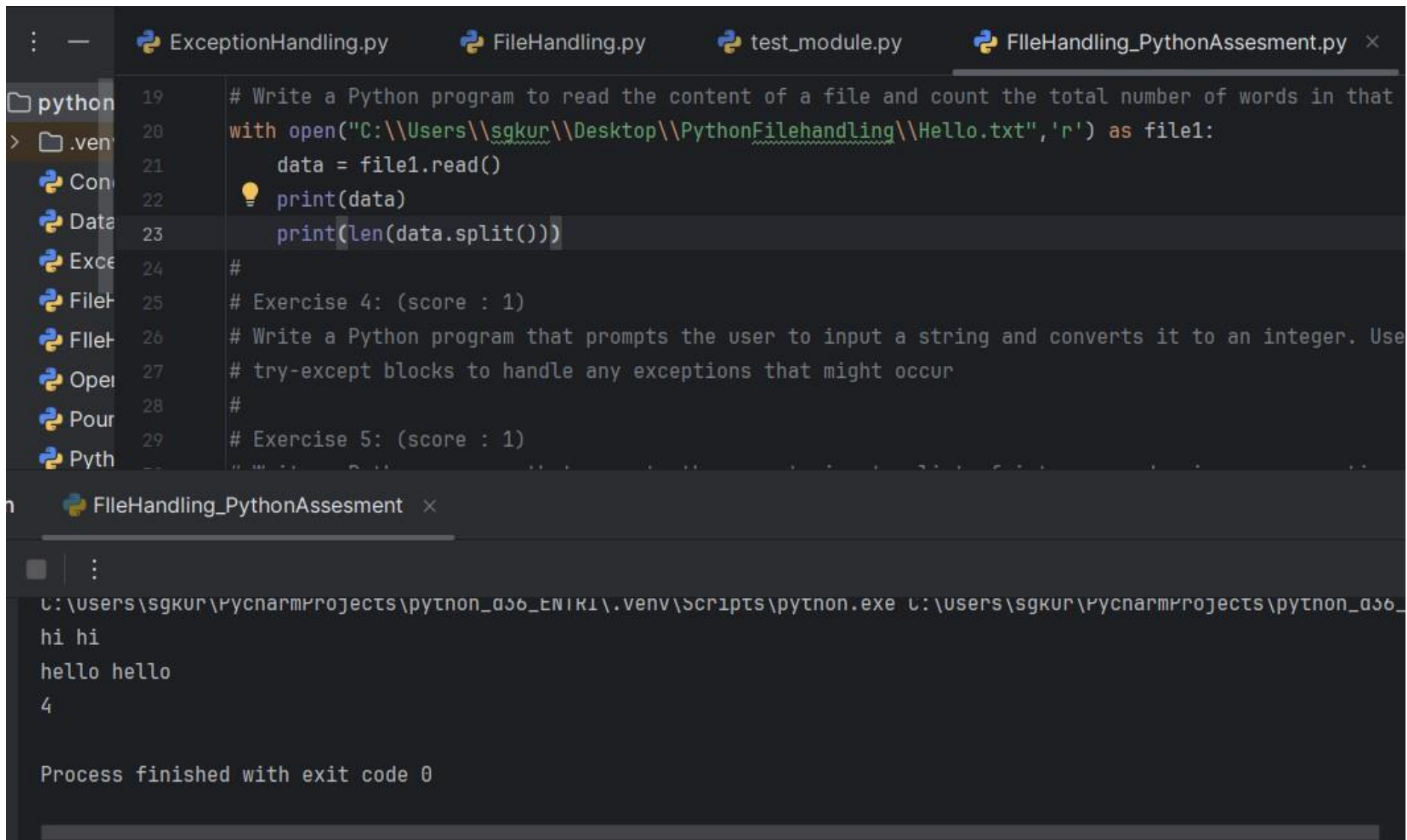
```
ExceptionHandling.py  FileHandling.py  test_module.py  FileHandling_PythonAssesment.py x
python
1  # Exercise 1: (score : 1)
2  # Write a Python program to read a file and display its contents
3  #file1=open("C:\\Users\\sgkur\\PycharmProjects\\python_d36_ENTRI\\Sample.txt",'r')
4  #print(file1)
5  #
6  # Exercise 2: (score : 1)
7  # Write a Python program to copy the contents of one file to another file
8  def copy_new_file(source, destination): 1 usage
9      with open("C:\\Users\\sgkur\\PycharmProjects\\python_d36_ENTRI\\Sample.txt", 'r') as source_file:
10         with open("C:\\Users\\sgkur\\Desktop\\PythonFilehandling\\Hello.txt", 'w') as desti_file:
11             desti_file.write(source_file.read())
12

FileHandling_PythonAssesment x
C:\Users\sgkur\PycharmProjects\python_d36_ENTRI\venv\Scripts\python.exe C:\Users\sgkur\PycharmProjects\python_d36_ENTRI\FileH
Process finished with exit code 0
```

Q.2



Q.3



The image shows a Python IDE with a dark theme. The top panel displays four open files: `ExceptionHandling.py`, `FileHandling.py`, `test_module.py`, and `FileHandling_PythonAssesment.py`. The `FileHandling_PythonAssesment.py` file is active, showing the following code:

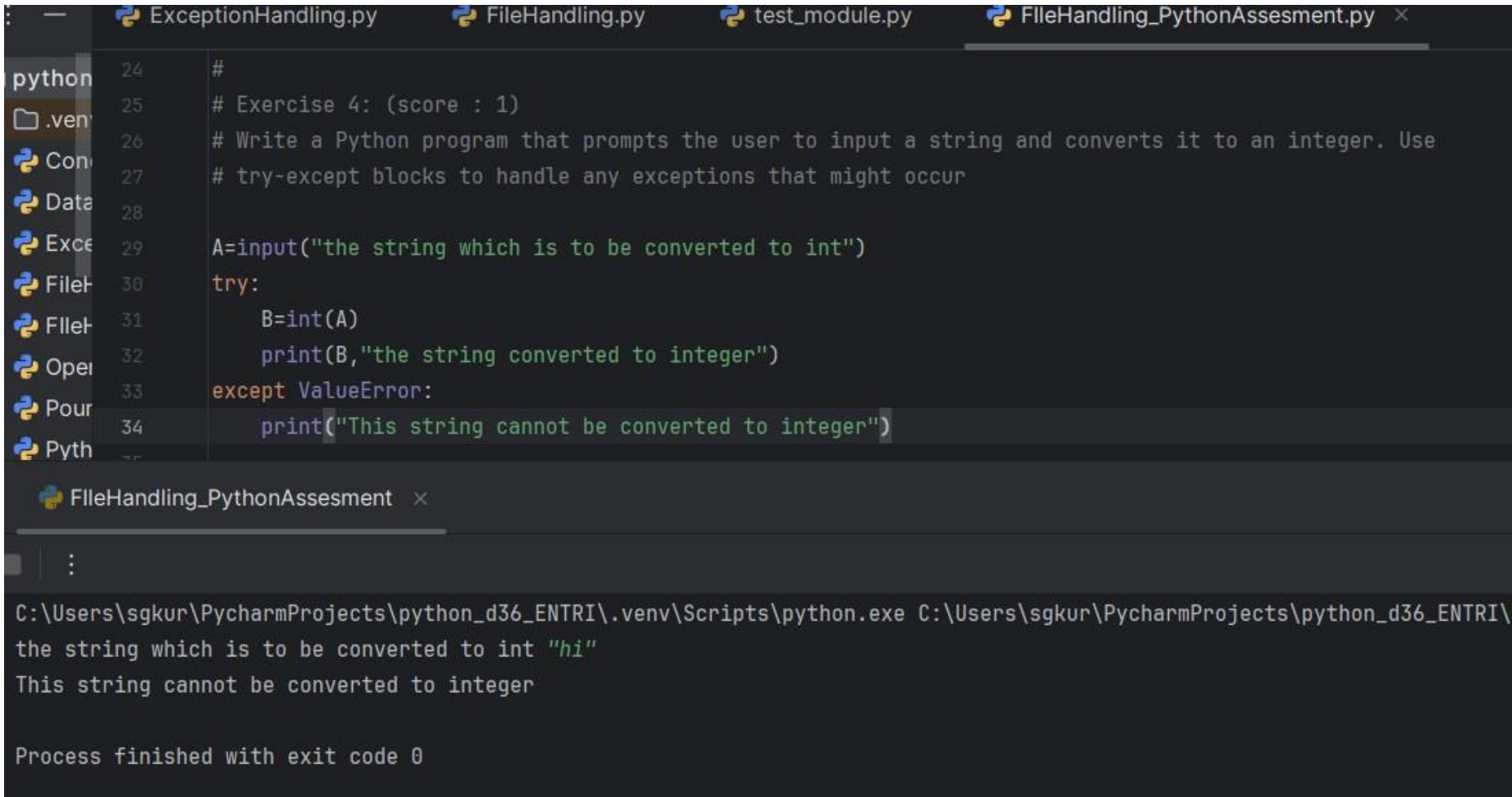
```
19 # Write a Python program to read the content of a file and count the total number of words in that
20 with open("C:\\Users\\sgkur\\Desktop\\PythonFilehandling\\Hello.txt", 'r') as file1:
21     data = file1.read()
22     print(data)
23     print(len(data.split()))
24
25 # Exercise 4: (score : 1)
26 # Write a Python program that prompts the user to input a string and converts it to an integer. Use
27 # try-except blocks to handle any exceptions that might occur
28 #
29 # Exercise 5: (score : 1)
```

The bottom panel shows the output of the program, indicating the file path, the content of the file, and the word count.

```
C:\Users\sgkur\PycharmProjects\python_d36_EWIKI\venv\Scripts\python.exe C:\Users\sgkur\PycharmProjects\python_d36_
hi hi
hello hello
4

Process finished with exit code 0
```

Q.4



```
python 24 #
      25 # Exercise 4: (score : 1)
      26 # Write a Python program that prompts the user to input a string and converts it to an integer. Use
      27 # try-except blocks to handle any exceptions that might occur
      28
      29 A=input("the string which is to be converted to int")
      30 try:
      31     B=int(A)
      32     print(B,"the string converted to integer")
      33 except ValueError:
      34     print("This string cannot be converted to integer")

FileHandling_PythonAssesment x
:
C:\Users\sgkur\PycharmProjects\python_d36_ENTRI\.venv\Scripts\python.exe C:\Users\sgkur\PycharmProjects\python_d36_ENTRI\
the string which is to be converted to int "hi"
This string cannot be converted to integer

Process finished with exit code 0
```

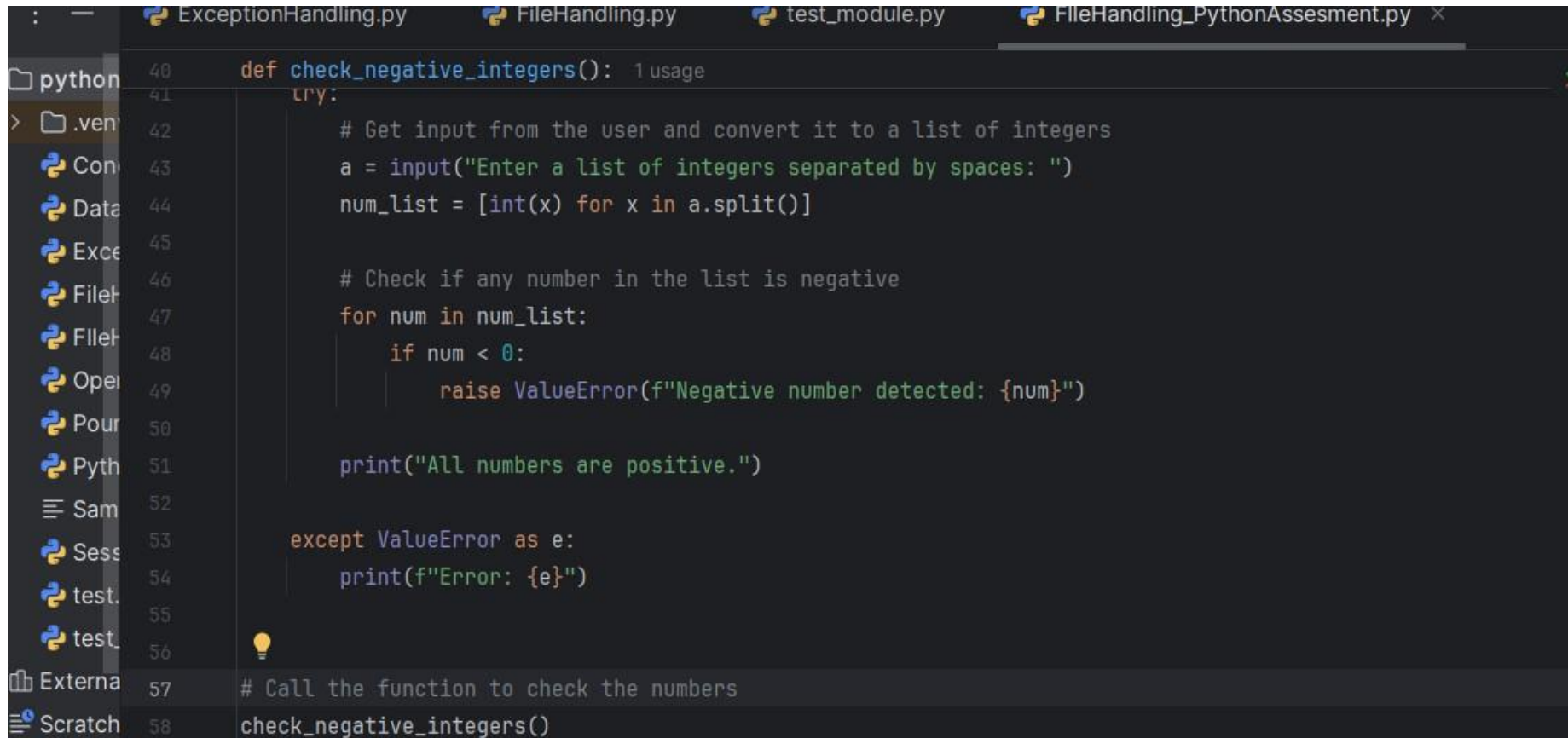
Q.4

```
python 24 #
        25 # Exercise 4: (score : 1)
        26 # Write a Python program that prompts the user to input a string and converts it to an integer. Use
        27 # try-except blocks to handle any exceptions that might occur
        28
        29 A=input("the string which is to be converted to int")
        30 try:
        31     B=int(A)
        32     print(B,"the string converted to integer")
        33 except ValueError:
        34     print("This string cannot be converted to integer")

FileHandling_PythonAssesment x
:
C:\Users\sgkur\PycharmProjects\python_d36_ENTRI\.venv\Scripts\python.exe C:\Users\sgkur\PycharmProjects\python_d36_ENTRI\
the string which is to be converted to int56
56 the string converted to integer

Process finished with exit code 0
```

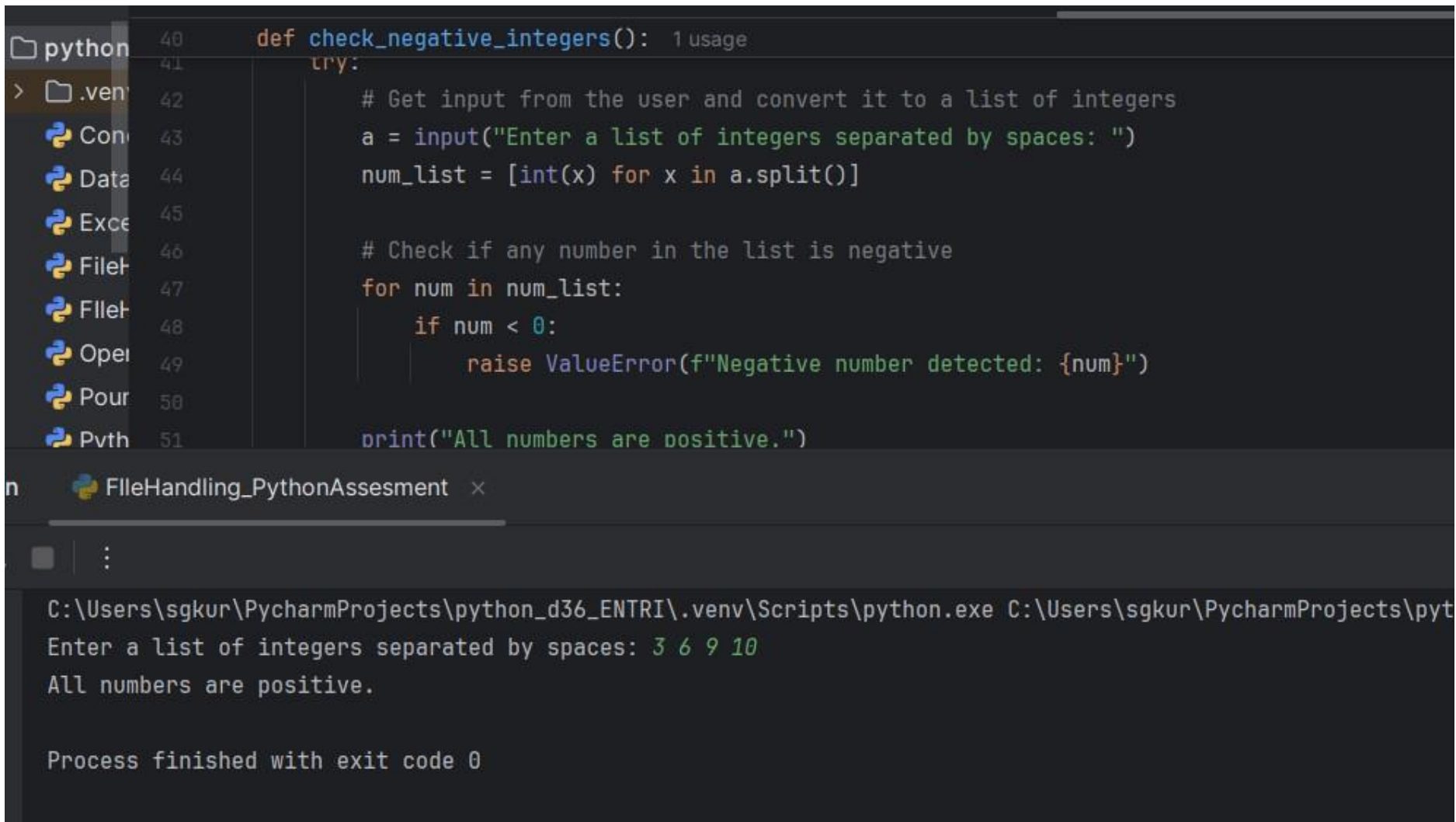

Q.5



The screenshot shows a Python IDE with a dark theme. The left sidebar displays a file explorer with a tree view containing folders like 'python', '.venv', and 'Extern', and files like 'Con', 'Data', 'Exce', 'FileH', 'FileH', 'Ope', 'Pour', 'Pyth', 'Sam', 'Sess', 'test.', 'test.', and 'Scratch'. The main editor window has tabs for 'ExceptionHandling.py', 'FileHandling.py', 'test_module.py', and 'FileHandling_PythonAssesment.py'. The active file is 'ExceptionHandling.py', which contains the following Python code:

```
40 def check_negative_integers(): 1 usage
41     try:
42         # Get input from the user and convert it to a list of integers
43         a = input("Enter a list of integers separated by spaces: ")
44         num_list = [int(x) for x in a.split()]
45
46         # Check if any number in the list is negative
47         for num in num_list:
48             if num < 0:
49                 raise ValueError(f"Negative number detected: {num}")
50
51         print("All numbers are positive.")
52
53     except ValueError as e:
54         print(f"Error: {e}")
55
56     # Call the function to check the numbers
57     check_negative_integers()
```

Q.5



The image shows a PyCharm IDE window with a Python file named `python_d36_ENTRI.py`. The code defines a function `check_negative_integers()` that prompts the user for a list of integers and checks if any are negative. The execution output shows the user inputting `3 6 9 10`, resulting in the message `All numbers are positive.`

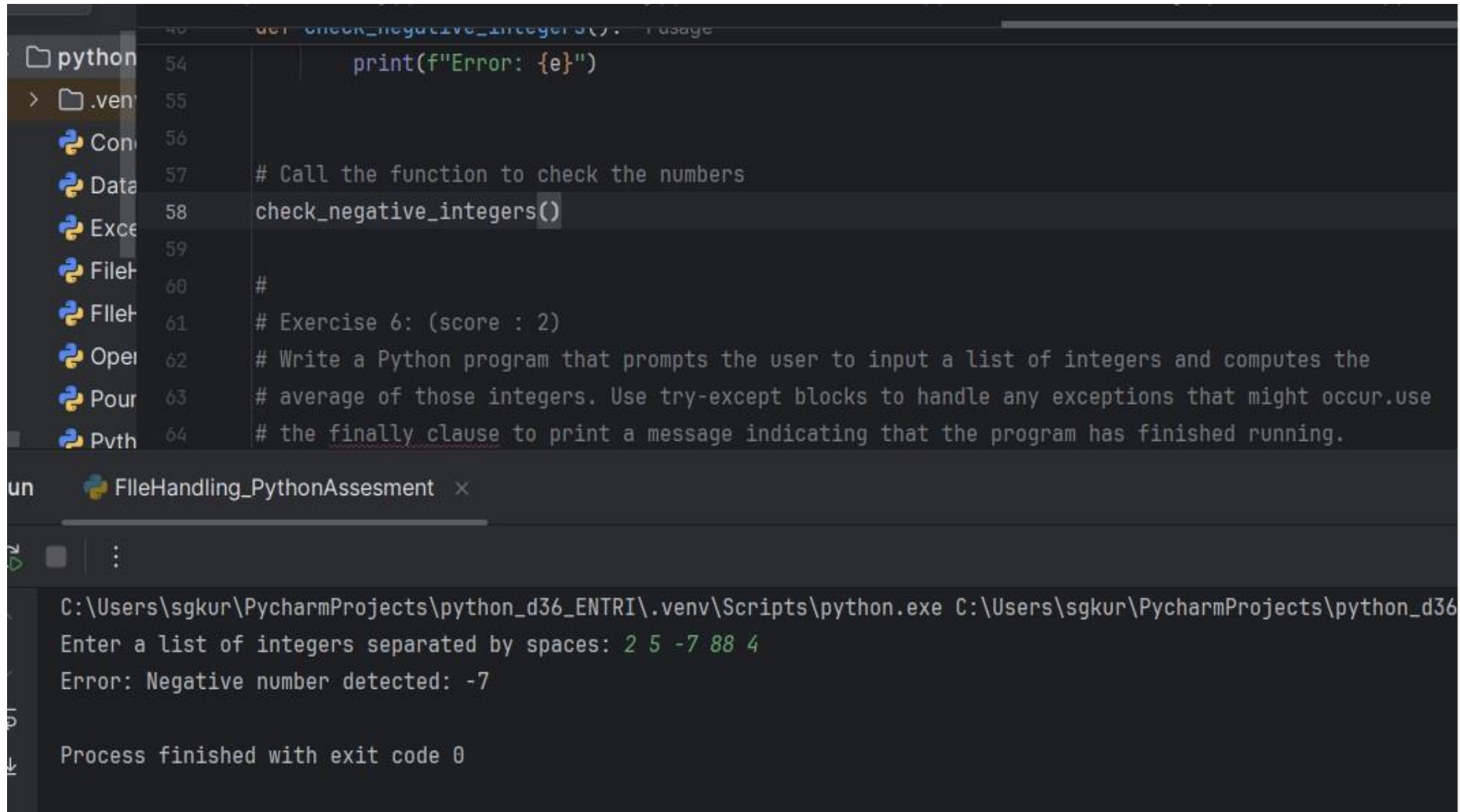
```
40 def check_negative_integers(): 1 usage
41     try:
42         # Get input from the user and convert it to a list of integers
43         a = input("Enter a list of integers separated by spaces: ")
44         num_list = [int(x) for x in a.split()]
45
46         # Check if any number in the list is negative
47         for num in num_list:
48             if num < 0:
49                 raise ValueError(f"Negative number detected: {num}")
50
51         print("All numbers are positive.")
```

FileHandling_PythonAssesment

C:\Users\sgkur\PycharmProjects\python_d36_ENTRI\.venv\Scripts\python.exe C:\Users\sgkur\PycharmProjects\pyt
Enter a list of integers separated by spaces: 3 6 9 10
All numbers are positive.

Process finished with exit code 0

Q.5



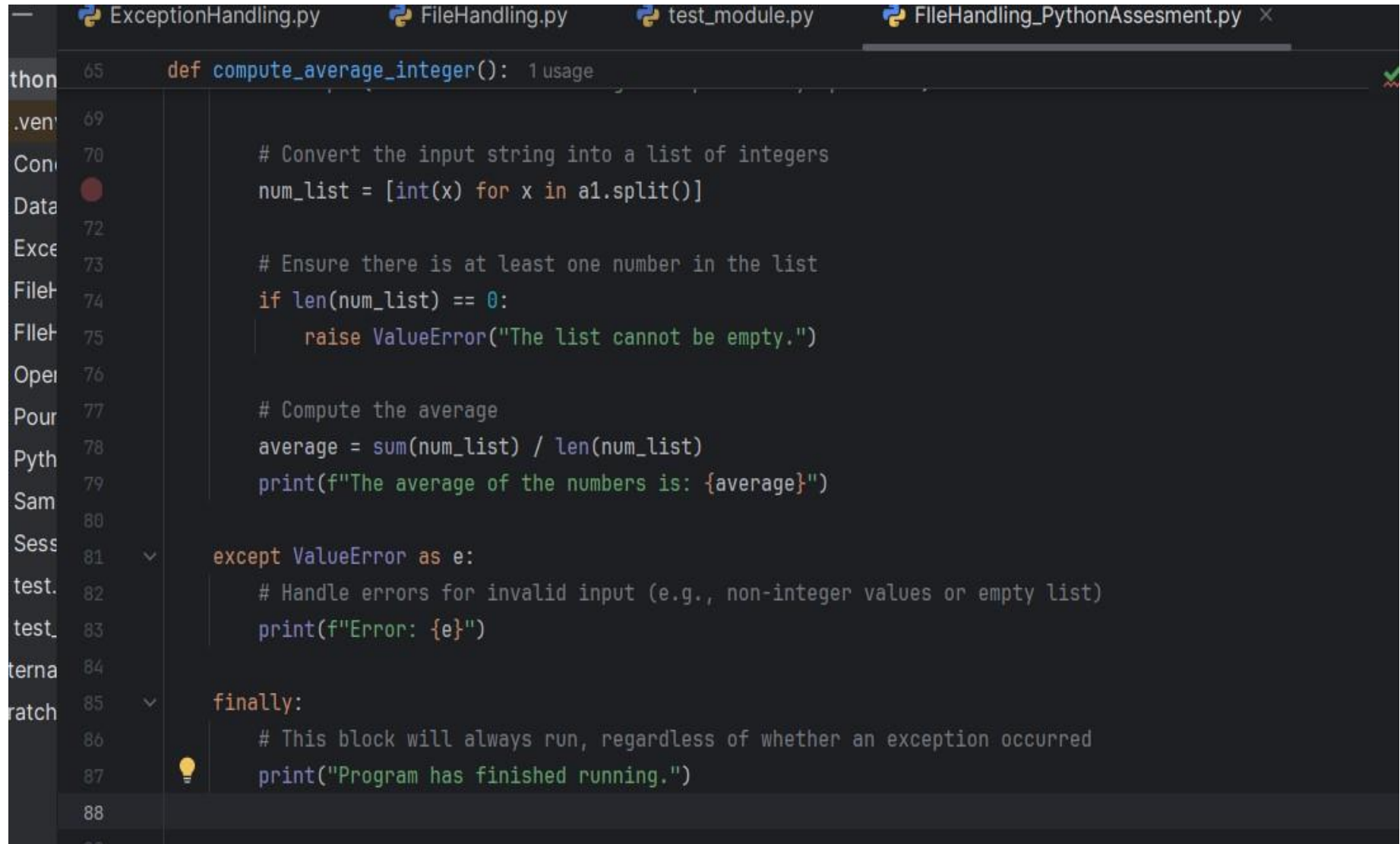
The screenshot displays the PyCharm IDE interface. The left sidebar shows a project structure with folders named 'python' and '.venv'. The main editor window shows a Python file named 'FileHandling_PythonAssesment'. The code in the editor is as follows:

```
40 def check_negative_integers():  
41     # Write a Python program that prompts the user to input a list of integers and computes the  
42     # average of those integers. Use try-except blocks to handle any exceptions that might occur. use  
43     # the finally clause to print a message indicating that the program has finished running.  
44     try:  
45         # Call the function to check the numbers  
46         check_negative_integers()  
47     except:  
48         print(f"Error: {e}")  
49     finally:  
50         # Exercise 6: (score : 2)  
51         # Write a Python program that prompts the user to input a list of integers and computes the  
52         # average of those integers. Use try-except blocks to handle any exceptions that might occur. use  
53         # the finally clause to print a message indicating that the program has finished running.
```

Below the editor, the 'Run' toolbar is visible, showing a green play button and a terminal icon. The terminal window at the bottom shows the execution of the program:

```
C:\Users\sgkur\PycharmProjects\python_d36_ENTRI\.venv\Scripts\python.exe C:\Users\sgkur\PycharmProjects\python_d36  
Enter a list of integers separated by spaces: 2 5 -7 88 4  
Error: Negative number detected: -7  
  
Process finished with exit code 0
```

Q.6



```
ExceptionHandling.py  FileHandling.py  test_module.py  FileHandling_PythonAssesment.py ×
65  def compute_average_integer(): 1 usage
69
70      # Convert the input string into a list of integers
71      num_list = [int(x) for x in a1.split()]
72
73      # Ensure there is at least one number in the list
74      if len(num_list) == 0:
75          raise ValueError("The list cannot be empty.")
76
77      # Compute the average
78      average = sum(num_list) / len(num_list)
79      print(f"The average of the numbers is: {average}")
80
81  except ValueError as e:
82      # Handle errors for invalid input (e.g., non-integer values or empty list)
83      print(f"Error: {e}")
84
85  finally:
86      # This block will always run, regardless of whether an exception occurred
87      print("Program has finished running.")
88
```

Q.6

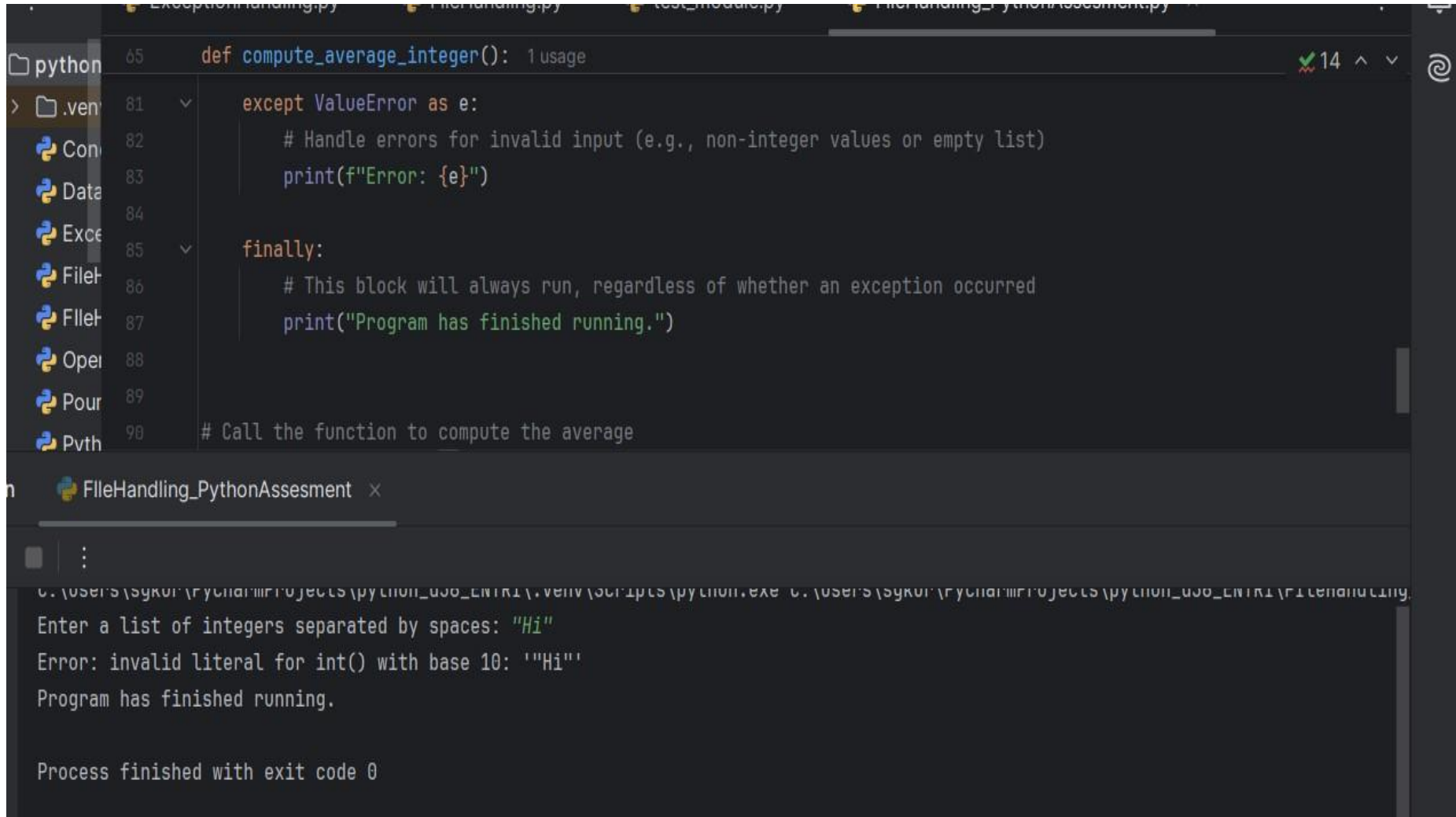
```
ExceptionHandling.py FileHandling.py test_module.py FileHandling_PythonAssesment.py x
python
> .venv
Con
Data
Exce
File
File
Ope
Pour
Pvth
87     print("Program has finished running.")
88
89
90     # Call the function to compute the average
91     compute_average_integer()
92
93     # Exercise 7 : (score : 2)
94     # Write a Python program that prompts the user to input a filename and writes a string to that file.
95     # Use try-except blocks to handle any exceptions that might occur and print a welcome message
96     # if there is no exception occurred.
```

Run FileHandling_PythonAssesment x

```
C:\Users\sykur\python\projects\python_000_entr1\.venv\scripts\python.exe C:\Users\sykur\python\projects\python_000_entr1\FileHandling_PythonAssesment.py
Enter a list of integers separated by spaces: 34 22 45 12
The average of the numbers is: 28.25
Program has finished running.

Process finished with exit code 0
```

Q.6



The screenshot displays a Python IDE with a file named `FileHandling_PythonAssesment.py`. The code defines a function `compute_average_integer()` that uses a `try-except-finally` block to handle `ValueError` exceptions. The `finally` block ensures that a message is printed regardless of whether an exception occurred. The function is then called to compute the average.

```
65 def compute_average_integer(): 1 usage
81     except ValueError as e:
82         # Handle errors for invalid input (e.g., non-integer values or empty list)
83         print(f"Error: {e}")
84
85     finally:
86         # This block will always run, regardless of whether an exception occurred
87         print("Program has finished running.")
88
89
90 # Call the function to compute the average
```

The terminal output shows the program's execution. It prompts the user to enter a list of integers, but the input `"Hi"` is invalid, resulting in a `ValueError`. The program then prints the message `Program has finished running.` and exits with a code of 0.

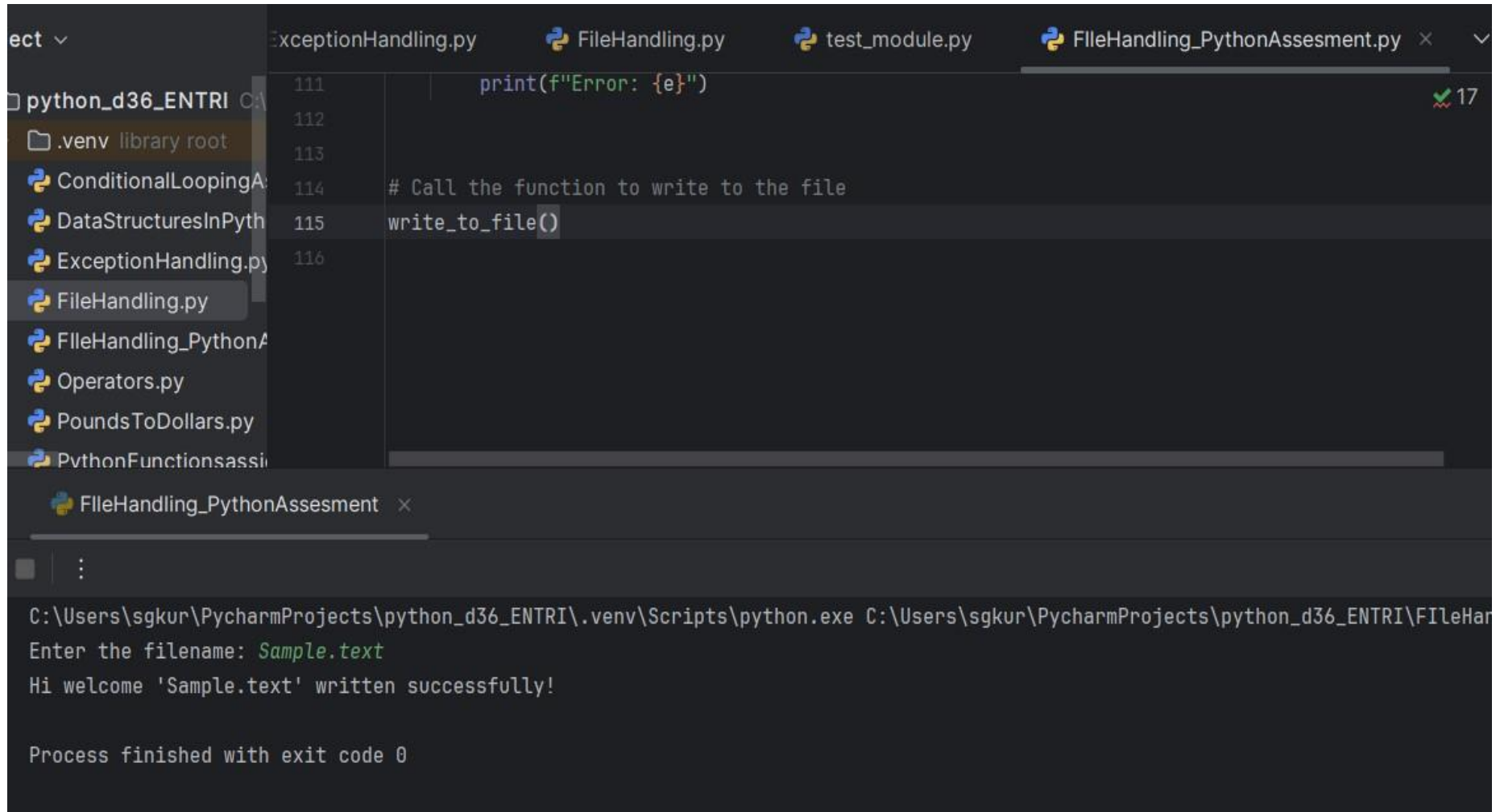
```
C:\Users\Sykur\PycharmProjects\python_assesment\venv\Scripts\python.exe C:\Users\Sykur\PycharmProjects\python_assesment\FileHandling_PythonAssesment.py
Enter a list of integers separated by spaces: "Hi"
Error: invalid literal for int() with base 10: '"Hi"'
Program has finished running.

Process finished with exit code 0
```


Q.7

```
93 # Exercise 7 : (score : 2) ✔ 17 ^ v
94 # Write a Python program that prompts the user to input a filename and writes a string to that file.
95 # Use try-except blocks to handle any exceptions that might occur and print a welcome message
96 # if there is no exception occurred.
97 def write_to_file(): # usage
98     try:
99         # Prompt the user for the filename
100         filename = input("Enter the filename: ")
101
102         # Open the file in write mode and write a string to it
103         with open("C:\\Users\\sgkur\\PycharmProjects\\python_d36_ENTRI\\Sample.txt", 'w') as file:
104             file.write("This is a sample string written to the file.")
105
106         # Print a welcome message if no exceptions occur
107         print(f"Hi welcome '{filename}' written successfully!")
108
109     except Exception as e:
110         # Handle any exceptions that occur (e.g., file permission issues)
111         print(f"Error: {e}")
112
113
114 # Call the function to write to the file
```

Q.7

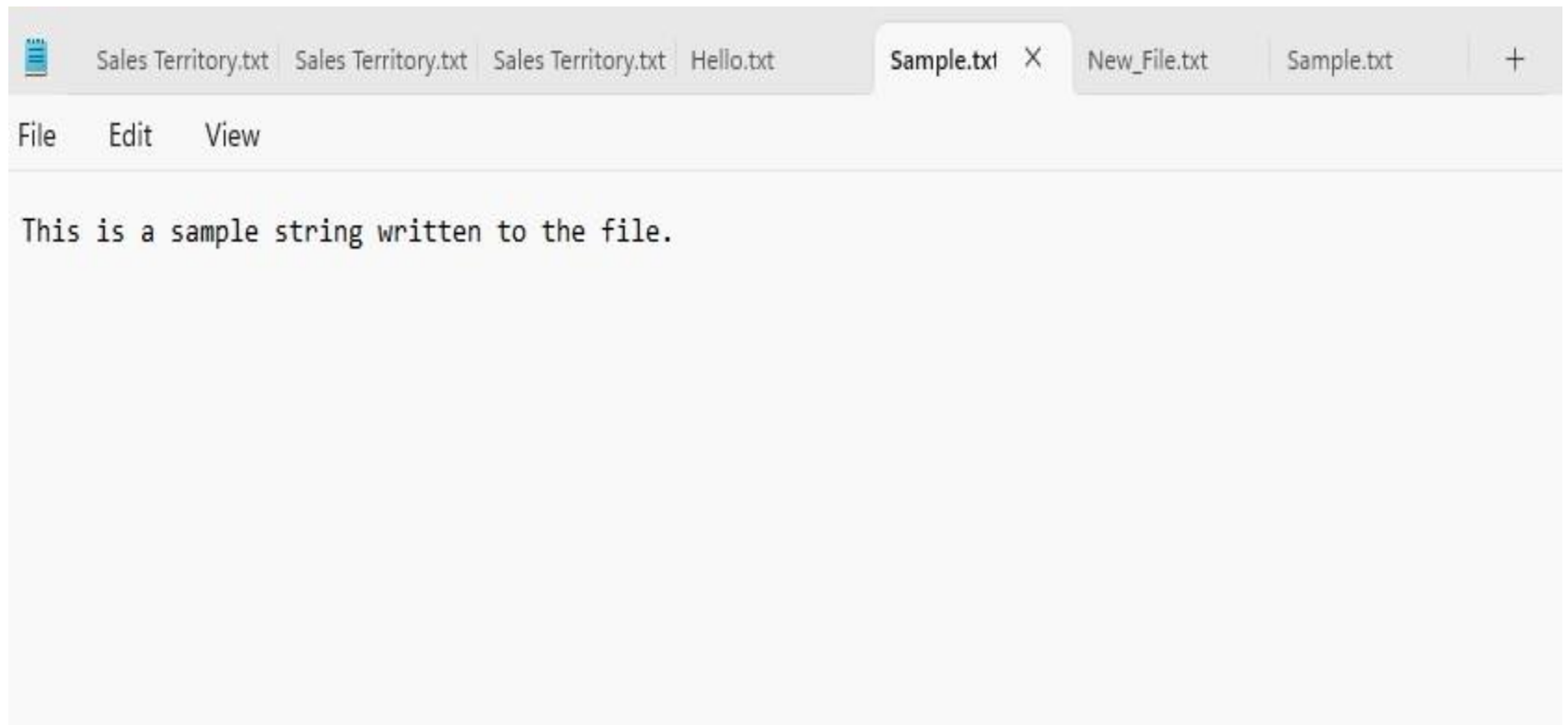


```
ExceptionHandling.py  FileHandling.py  test_module.py  FileHandling_PythonAssesment.py x
python_d36_ENTRI 111 print(f"Error: {e}")
                  112
                  113
                  114 # Call the function to write to the file
                  115 write_to_file()
                  116

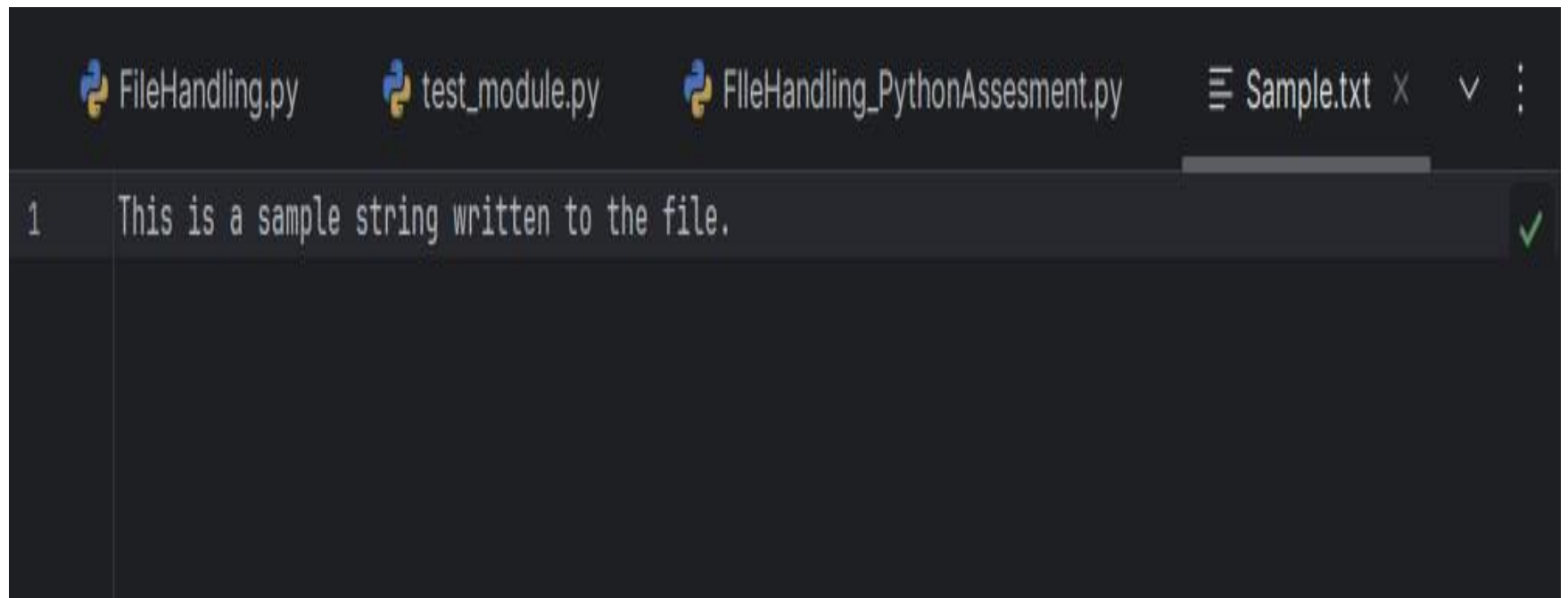
FileHandling_PythonAssesment x
C:\Users\sgkur\PycharmProjects\python_d36_ENTRI\.venv\Scripts\python.exe C:\Users\sgkur\PycharmProjects\python_d36_ENTRI\FILEHar
Enter the filename: Sample.text
Hi welcome 'Sample.text' written successfully!

Process finished with exit code 0
```


Q.7



Q.7



The image shows a code editor interface with a dark theme. At the top, there are four tabs: 'FileHandling.py', 'test_module.py', 'FileHandling_PythonAssesment.py', and 'Sample.txt'. The 'Sample.txt' tab is active and highlighted. Below the tabs, the editor displays a single line of text: '1 This is a sample string written to the file.' The line number '1' is in the left margin. A green checkmark is visible at the end of the line, indicating successful execution or completion. The background is dark gray, and the text is light gray.